**Câu 1 :** đã cài đặt

**Câu 2:**

Git là một hệ thống quản lý phiên bản mã nguồn phổ biến, trong khi GitHub là một dịch vụ lưu trữ mã nguồn trực tuyến sử dụng Git. Dưới đây là một số lợi ích của Git và GitHub:

Lợi ích của Git

Quản lý phiên bản: Git cho phép bạn lưu trữ nhiều phiên bản của mã nguồn, để bạn có thể truy xuất lại các phiên bản cũ nếu cần thiết.

Quản lý nhánh: Git cho phép bạn tạo và quản lý các nhánh khác nhau của mã nguồn, để bạn có thể phát triển các tính năng mới hoặc sửa lỗi mà không ảnh hưởng đến mã nguồn gốc.

Hợp nhất mã nguồn: Git cho phép bạn hợp nhất các nhánh khác nhau của mã nguồn, để bạn có thể đưa các tính năng mới vào mã nguồn chính một cách an toàn.

Truy cập nhanh: Git cho phép bạn truy cập nhanh chóng các phiên bản và nhánh khác nhau của mã nguồn.

Lợi ích của GitHub:

Lưu trữ mã nguồn trực tuyến: GitHub cho phép bạn lưu trữ mã nguồn trực tuyến, để bạn có thể truy cập nó từ bất kỳ đâu trên thế giới

Quản lý dự án: GitHub cung cấp các công cụ để quản lý dự án, bao gồm theo dõi các vấn đề, giao nhiệm vụ, và theo dõi tiến độ dự án.

Kết nối cộng đồng: GitHub là một cộng đồng lớn với hàng triệu lập trình viên trên toàn thế giới, bạn có thể kết nối với họ để học hỏi hoặc đóng góp vào dự án của mình.

Điều khiển quyền truy cập: GitHub cho phép bạn kiểm soát quyền truy cập vào dự án của mình, để bạn có thể cho phép người khác đóng góp hoặc chỉ xem mã nguồn của bạn.

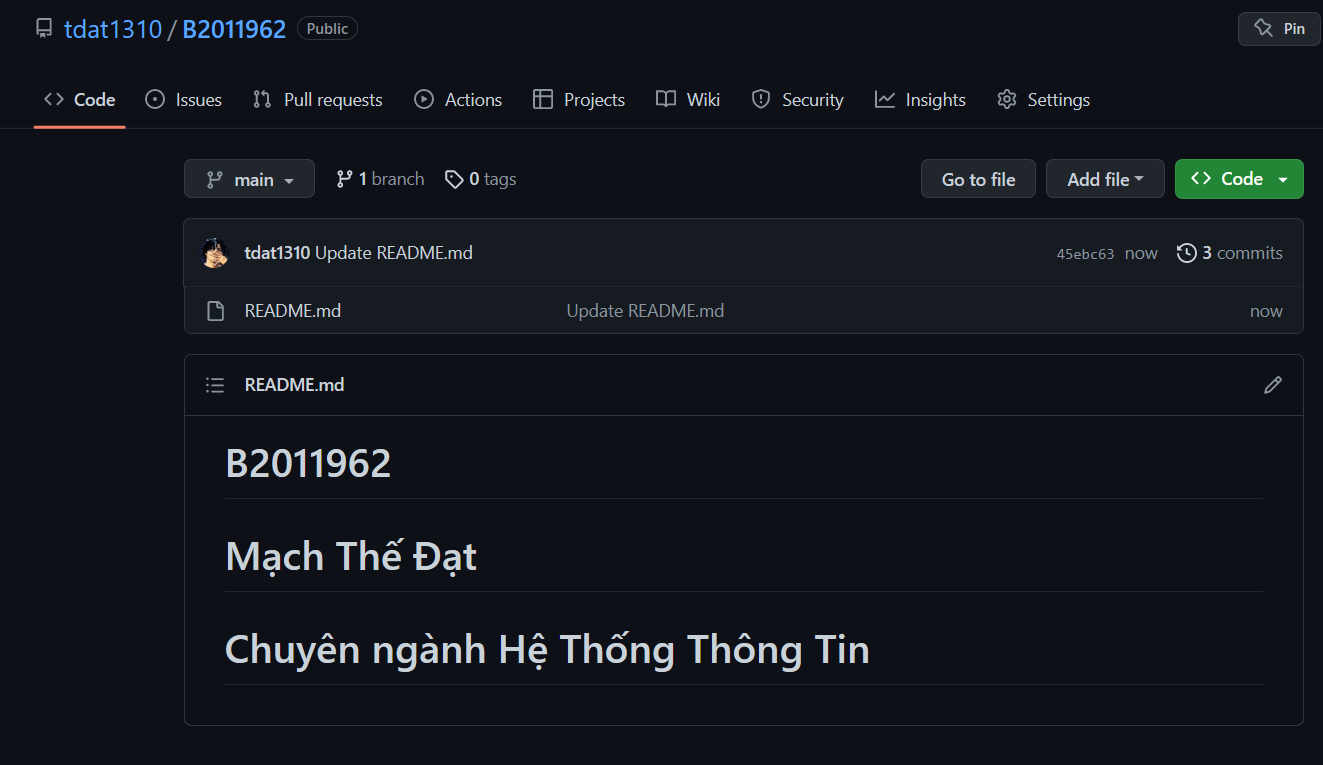
**Câu 3:**

Repositories trong GitHub là nơi lưu trữ mã nguồn của một dự án cụ thể. Mỗi repository bao gồm các tập tin và thư mục, lịch sử sửa đổi, các nhánh khác nhau của dự án, và các tính năng quản lý phiên bản khác.

Tùy vào gói đăng ký của bạn, GitHub cung cấp dung lượng lưu trữ khác nhau cho repository. Miễn phí, bạn có thể tạo và lưu trữ không giới hạn repository trên GitHub, nhưng tổng dung lượng lưu trữ của bạn sẽ bị giới hạn (tính từ tất cả các repository và các tập tin liên quan đến tài khoản GitHub của bạn). Nếu bạn muốn nâng cấp gói đăng ký của mình, bạn có thể mua các gói có dung lượng lưu trữ lớn hơn.

Về kích thước tập tin, GitHub cho phép bạn tải lên các tập tin lên đến 100MB. Tuy nhiên, nếu bạn muốn tải lên các tập tin lớn hơn, bạn có thể sử dụng các dịch vụ lưu trữ khác như Git Large File Storage (Git LFS) để quản lý các tệp lớn hơn trong repository của mình

Trang readme đã tạo :



Đường link repository : [tdat1310/B2011962 (github.com)](https://github.com/tdat1310/B2011962)

**Câu 4:**

Branch trong Git là một phiên bản riêng của mã nguồn trong một repository. Nó cho phép người dùng tách nhánh (branch) của mã nguồn hiện tại và tiến hành phát triển, thêm tính năng mới, sửa lỗi hoặc thực hiện bất kỳ thay đổi nào khác mà không ảnh hưởng đến nhánh chính của mã nguồn.

Khi tạo một branch mới, Git sẽ tạo ra một bản sao của nhánh hiện tại và cho phép người dùng thực hiện các thay đổi trên branch mới. Khi các thay đổi được thực hiện, người dùng có thể chọn hợp nhất (merge) nhánh mới với nhánh chính hoặc nhánh khác để đưa các thay đổi mới vào nhánh chính.

Tạo Branch tên “nhanhphu-<mssv của bạn>” :



**Câu 5:**

Commit trong Git là hành động lưu lại các thay đổi đã thực hiện trên repository. Nó tạo ra một phiên bản (version) mới của mã nguồn trong repository với các thay đổi đã được lưu lại.

Push trong Git là hành động đẩy (push) các thay đổi đã commit lên repository trên máy chủ từ máy tính của người dùng. Sau khi commit các thay đổi, người dùng cần push chúng lên để cập nhật repository trên máy chủ và cho phép các thành viên khác trong nhóm cập nhật các thay đổi này.

Pull trong Git là hành động kéo (pull) các thay đổi mới nhất từ repository trên máy chủ về máy tính của người dùng. Nó cho phép người dùng cập nhật các thay đổi mới nhất và tích hợp chúng vào repository trên máy tính của mình.

Khi làm việc với Git, người dùng thường thực hiện chu trình thay đổi - commit - push để cập nhật repository trên máy chủ và chu trình pull - thay đổi - commit để cập nhật mã nguồn trên máy tính cá nhân.

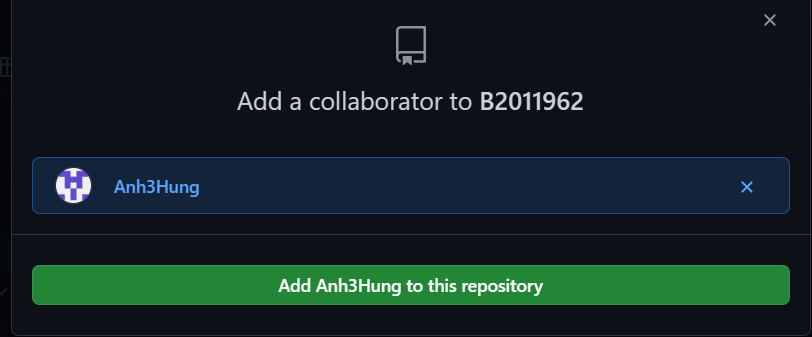
**Câu 8:**

Các lệnh trên thực hiện các hành động cơ bản trong Git để lưu trữ và quản lý mã nguồn:

git add .: Lệnh này sẽ thêm tất cả các thay đổi đã được chỉnh sửa hoặc tạo mới từ thư mục làm việc của bạn (working directory) vào vùng đệm (staging area) để chuẩn bị cho việc commit. Dấu chấm "." ở đây cho biết tất cả các tệp tin và thư mục cần được thêm vào.

git commit -m "upload du lieu": Lệnh này tạo ra một phiên bản mới của mã nguồn dựa trên các thay đổi đã được thêm vào vùng đệm bằng lệnh git add. Thông điệp commit ("upload du lieu") cho biết mục đích của commit này, ví dụ như một mô tả ngắn gọn về những gì đã thay đổi.

git push: Lệnh này sẽ đẩy các thay đổi của commit lên repository trên máy chủ của bạn. Các thay đổi này sẽ được lưu trữ và trở nên có sẵn cho các thành viên khác của nhóm cập nhật và tích hợp vào repository của họ

**Câu 10 :**

****